Finding Lane lines on the Road

Harishma Dayanidhi

September 1, 2017

1 The goals/steps of this project

- Make a pipeline that finds lane lines on the road
- Reflect on your work in a written report

2 Reflection

2.1 Description of the pipeline

My pipeline consisted of the following steps

- Read the image
- Convert the image to grayscale
- Perform canny edge detection
- Use region filters to find areas of interest
- Perform Hough transform to get all sets of lines
- Combine all the dotted lines on the left and right to form two continuous lines on left and right respectively.

Drawlines function implementation

First I calculated the slope for every line segment identified in the hough transform stage. Then I separated these lines into multiple bins based on the slopes (a similarity metric based on a threshold). Then for each bin the maximum and minimum (x,y) coordinates were calculated and in some cases extrapolated to the bottom of the image if they ended before. Among all the slopes identified, the lowest positive slope and the highest negative slope whose absolute were values are above a certain hand-tuned threshold were selected as the lane boundaries. The rationale behind this was that the lowest positive slope and the highest negative slope will belong to the lines closest to the camera frame.

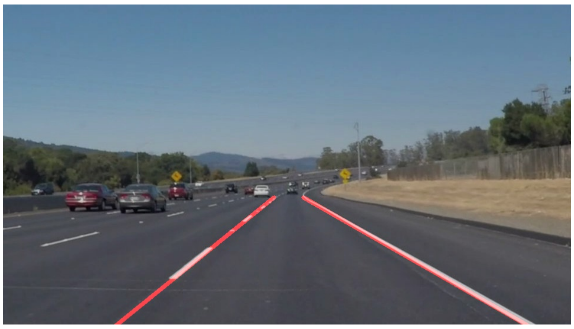

Figure 1: A sample final output with the lane lines drawn in red

2.2 Shortcomings with current pipeline

The current implementation does not have a robust way of dealing with curved lines. It approximates the slopes by picking a representative slope among all similar slopes but as can be seen in the challenge video it is not robust to lanes lines captured at certain angles.

2.3 Possible improvements to current pipeline

Instead of estimating the slope it might be better to average the slopes and always including the line segments near the edges of the image in the final line.